The impact of user impatience on Internet performance

Bert Zwart

February 16, 2006

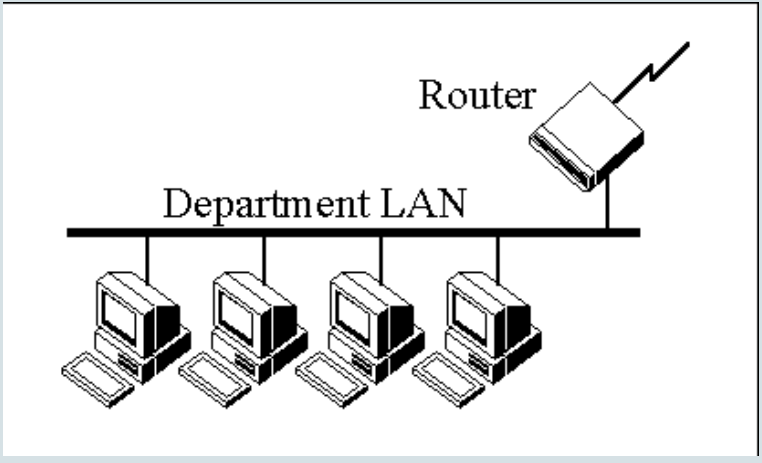
Joint work with Christian Gromoll (Stanford) and Philippe Robert (INRIA).

To appear in ACM Sigmetrics, 2006.

Impatience causes significant overhead

- Networks are often very congested ⇒ users receive poor service.
- Feldmann *et al.* (1999): 11 % of Internet data transfers are aborted prior to completion; these transfers correspond to 20 % of the total traffic.

Main questions addressed in this talk



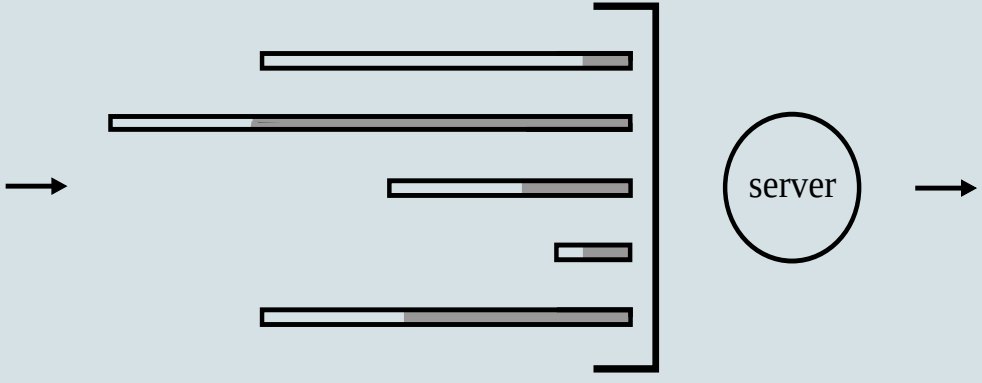
- What is the fraction of users that terminate their job before completion?
- How much bandwidth is wasted on such users?
- How can we limit the impact of impatience?

Overview of this talk

- Modeling impatience in bandwidth sharing networks.
- Performance Analysis.
- A remedy: Admission control.
- Reattempts.
- Summary.
- Related problems.

Modeling a bottleneck link in a bandwidth network

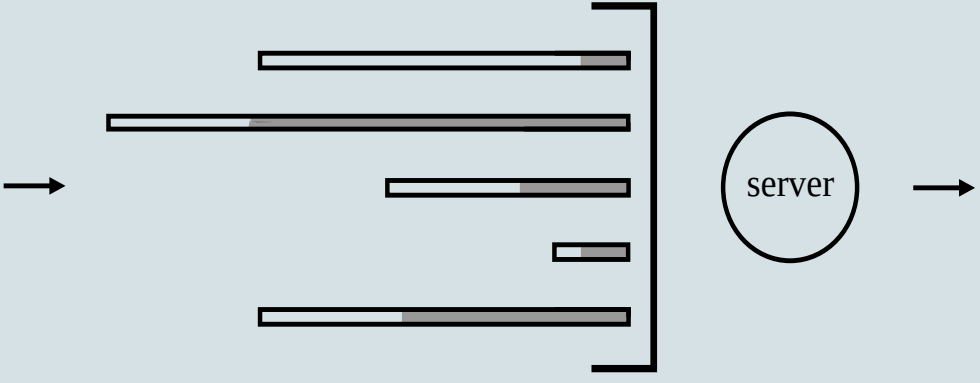
- Bandwidth sharing networks often use a variant of the TCP protocol.
- Crucial property of TCP: If n identical users share the network for a long time, they eventually receive the same service rate.
- Processor sharing (PS) is a service mechanism where the server serves all customers at equal speed.



PS is a flow-level model for TCP

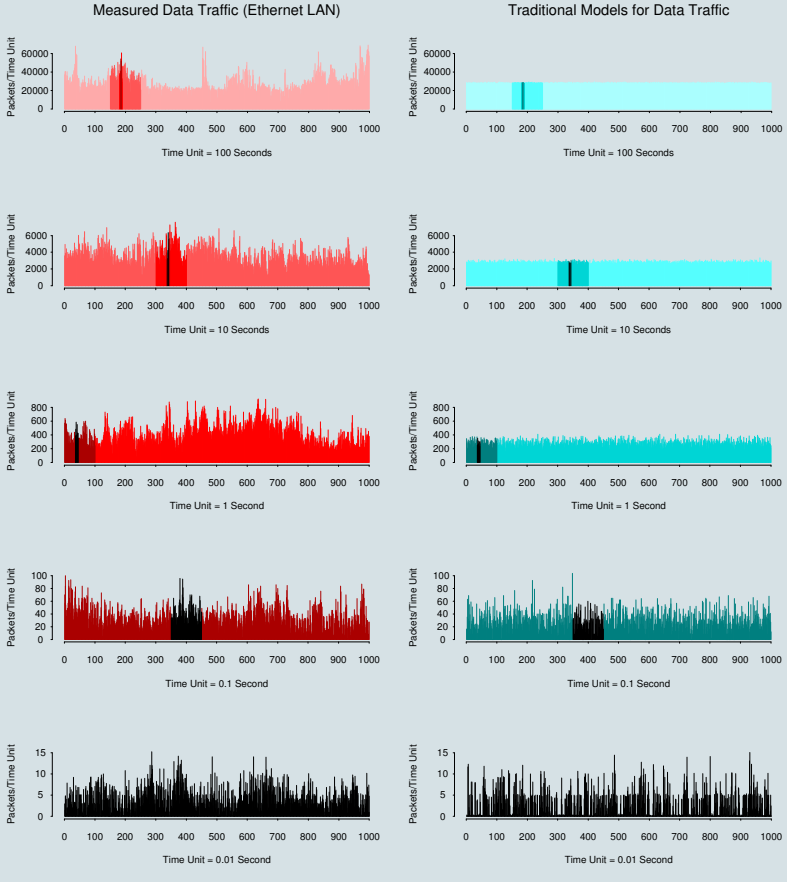
- Example: if there are 5 customers in the system, then each customer is served with rate $1/5$. When an extra customer enters, all customers will be *immediately* served with rate $1/6$.

- Unlike TCP, PS adapts the long-term service rate immediately to the new situation. Therefore, *PS is an idealized version of TCP*.



From now on, we approximate TCP with PS.

Challenge I: Traffic is bursty



- LAN traffic vs. traffic generated by conventional model.
- Traffic is bursty at wide range of time scales (from 10 milliseconds to 100 seconds).
- Explanation and well-established fact: **File sizes have infinite variance**.

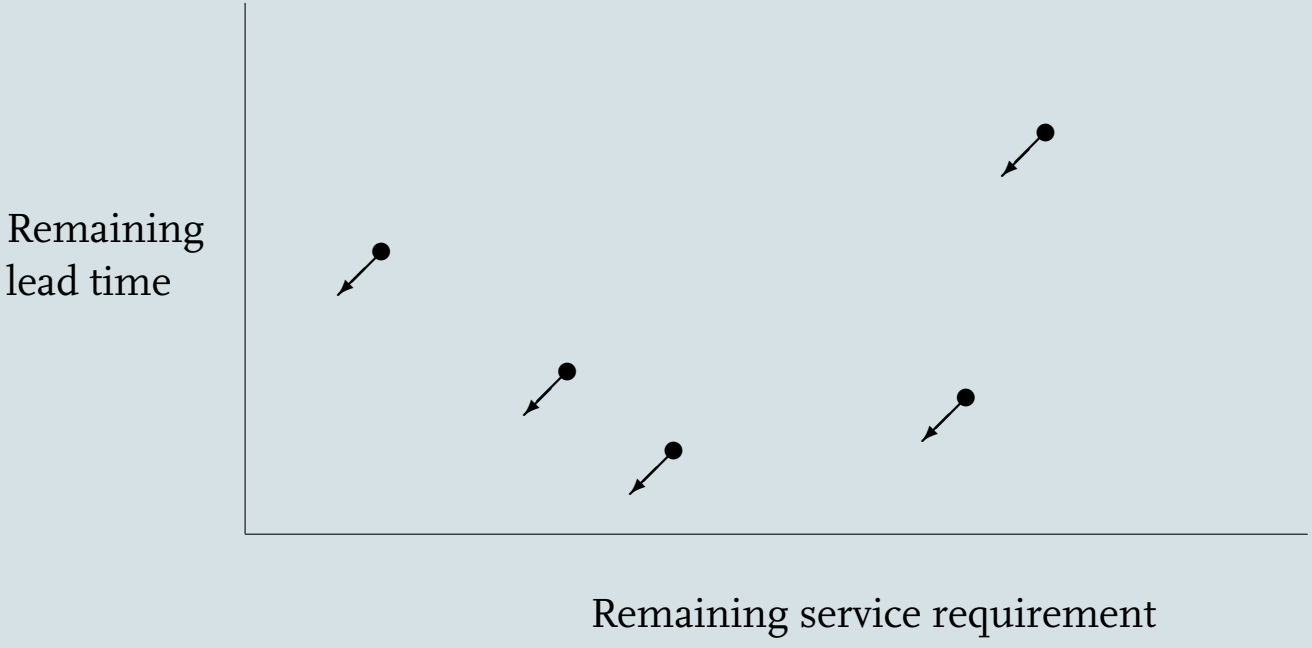
Challenge II: Huge gap in queueing literature

- The literature on FIFO queues with impatience is extensive, motivated by call center applications.
- There is hardly any literature on PS queues with impatience. Exceptions are Coffman *et al.* (1994) and Bonald & Roberts (2003).
- The reason is obvious. We encountered an exciting combination of three complicated features:
  - The system lacks memoryless properties.
  - The system is ***not*** work-conserving due to impatience.
  - Time-sharing allows customers to overtake: desirable, but intricate!

Processor sharing with impatience: the setup

- Users arrive at the system according to a renewal process with rate λ.
- Service requirements have a general distribution, which may have infinite variance.
- Each user has a lead time, which may be dependent on his service time.
- A user leaves due to impatience when his lead time expires.
- No upper bound on number of users simultaneously in the system.

Describing the model as a particle system



Snapshot of the system with 5 users. "Particles" move to the left with rate $1/5$ and downwards with rate 1.

Reducing model complexity by fluid scaling

- $Z(t)$: number of customers at time t.
- $Z(t), t \geq 0$ is a complicated non-Markov process.
- Therefore, we consider a fluid scaling. Informally, we scale time and space by a factor r, and replace the lead times D_i by rD_i.
- Interpretation: Server works at rate r, and customers arrive at rate λr.

Main convergence results

Assume that the system is overloaded: $\rho = \lambda \mathbf{E}[B] > 1$.

Theorem 1 (approximation of time-dependent behavior)
There exists a continuous function $z(\cdot)$ such that $\frac{1}{r} Z(rt) \to z(t)$.

Theorem 2 (approximation of steady-state behavior)
If $\rho > 1$ and also

$$\lambda \mathbf{E}[B 1_{\{D=\infty\}}] < 1, \qquad \mathbf{E}[\min\{B, D\}] < \infty,$$

then $z(t) \to z$ as $t \to \infty$, with z the positive solution of the equation

$$z = \lambda \mathbf{E}[\min\{zB, D\}].$$

Number of customers at time t

The process $z(\cdot)$ approximates the number of customers in the system.

$$z(t) = z_0 \mathbf{P}[B_0 > S(0,t), D_0 > t] + \lambda \int_0^t \mathbf{P}[B > S(s,t), D > t-s] \mathrm{d}s,$$

with

$$S(s,t) = \int_s^t \frac{1}{z(u)} \mathrm{d}u.$$

- $S(s,t)$ is the total service rate between time s and time t.
- $z_0 \mathbf{P}[B_0 > S(0,t), D_0 > t]$: total "mass" at time 0 which is still in the system at time t.
- $\mathbf{P}[B > S(s,t), D > t-s]$: fraction of mass arrived at time s which is still in system at time t.

Modeling impatience in TCP: Summary

- We approximated TCP by an idealized version: PS.
- PS with impatience is still too complicated to analyze.
- A fluid approximation reduced the random process to a fluid model.
- Steady-state is approximated by the simple fixed-point equation

$$z = \lambda \mathbf{E}[\min\{zB, D\}].$$

Overview

- Modeling impatience in TCP networks.
- ***Performance Analysis.***
- A remedy: Admission control.
- Reattempts.
- Summary.
- Related problems.

Interpretation of the fixed point equation

Let Z^r be the steady-state number of users.

Let V^r be the steady-state sojourn time of a user.

$V^r = \min\{V_p^r, rD\}$ with V_p^r the *potential* sojourn time (if the customer would not be impatient).

Little's law:

$$\mathbf{E}[Z^r] = \lambda \mathbf{E}[V^r] = \lambda \mathbf{E}[\min\{V_p^r, Dr\}].$$

What is V_p^r?

Combining Little's law and the snapshot principle

If the number of customers in the system is approximately constant during a customer's sojourn time as r becomes large, then

$$V_p^r = (Z^r + o(r))B.$$

This is called the **snapshot principle**: in equilibrium, a customer does not observe any fluctuations of the system during his sojourn.

Combined with Little's law, this gives:

$$\mathbf{E}[Z^r] = \lambda \mathbf{E}[\min\{(Z^r + o(r))B, rD\}].$$

Divide both sides by r and let $r \to \infty$ to get

$$z = \lambda \mathbf{E}[\min\{zB, D\}].$$

Performance measures

- Number of users in the system: rz, with

$$z = \lambda \mathbf{E}[\min\{zB, D\}].$$

- Fraction of users that do not renege: $P_s = \mathbf{P}[zB < D]$.

- Server utilization: $\rho_s = \lambda \mathbf{E}[B; zB < D]$.

- Time-dependent reneging rate $d(t)$.

Will it help to make customers more patient?

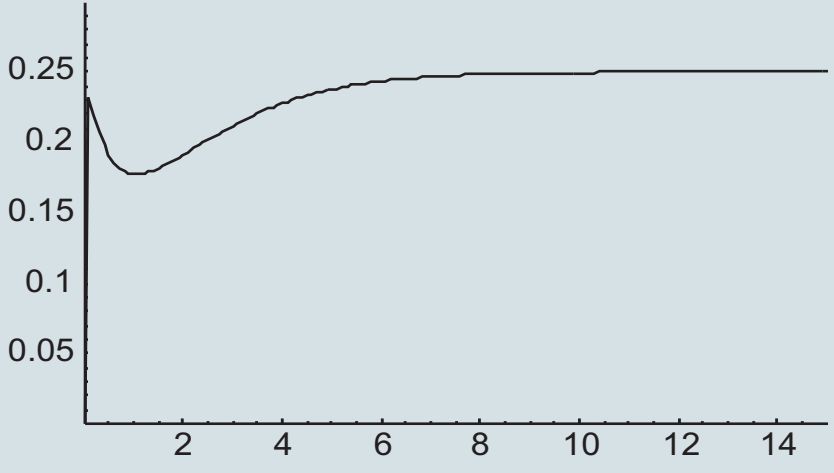
- Suppose that customers become twice as patient.
- How much will the fraction of successful customers P_s increase?

Answer: P_s will not increase at all!

Lesson: If the system is overloaded, the average lead-time is not important.

Making customers more patient helps temporarily

Time-dependent behavior of the reneging rate $d(t)$ for a system which is in equilibrium at time 0 and in which customers arriving after time 0 are twice as patient as before time 0. ($\lambda = 2$, $\mu = 1$, $\nu_0 = 2$, $\nu_1 = 1$)



$P_{s,new} = P_{s,old}$ – proof is quite simple

$$z_{old} = \lambda \mathbf{E}[\min\{z_{old}B, D\}].$$

$$2z_{old} = \lambda \mathbf{E}[\min\{2z_{old}B, 2D\}].$$

$$z_{new} = \lambda \mathbf{E}[\min\{z_{new}B, 2D\}].$$

$\Rightarrow z_{new} = 2z_{old}$

$$\begin{aligned} P_{s,new} &= \mathbf{P}[z_{new}B < 2D] \\ &= \mathbf{P}[2z_{old}B < 2D] \\ &= \mathbf{P}[z_{old}B < D] \\ &= P_{s,old}. \end{aligned}$$

Example 1: Linearly dependent lead times

Take $D = \Theta B$, with Θ and B independent.
Θ reflects the average service level expected by a customer.

If Θ is a constant θ (say), then

$$z = \rho \min\{\theta, z\}.$$

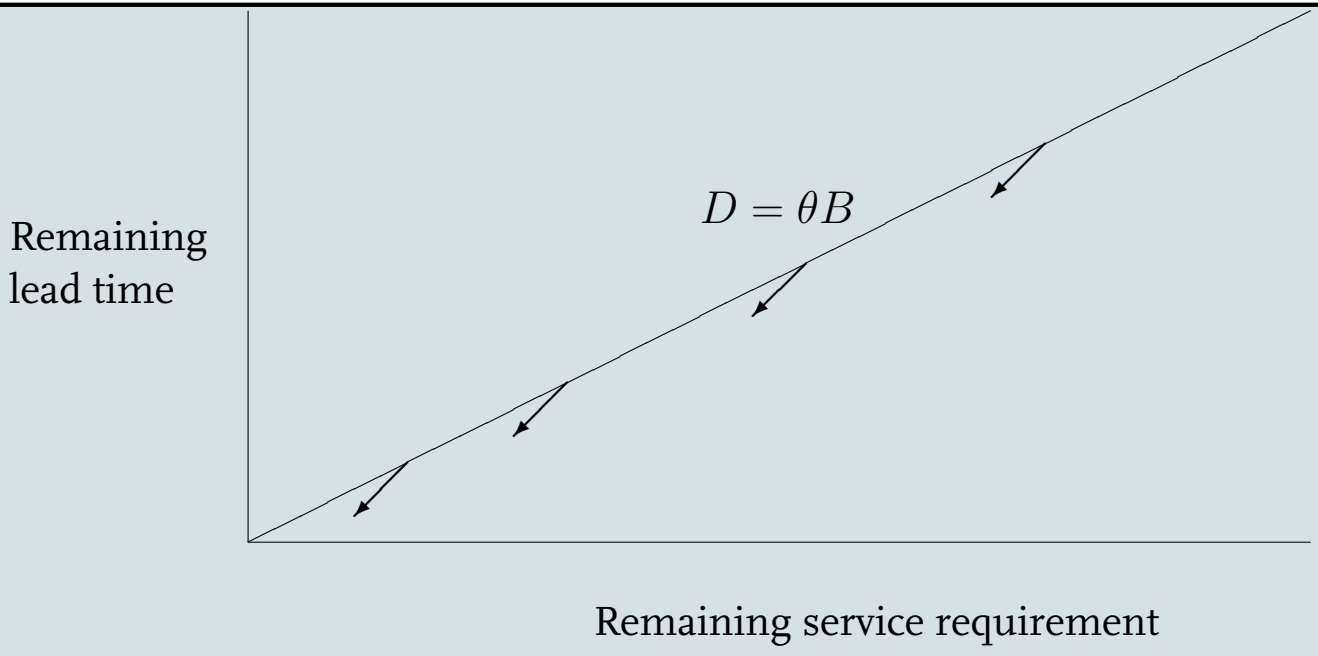
implying that $z = \rho\theta$.

Consequently:

$$P_s = \mathbf{P}[D > zB] = \mathbf{P}[\theta > z] = \mathbf{P}[\theta > \rho\theta] = 0.$$

All users in the system will be impatient!!

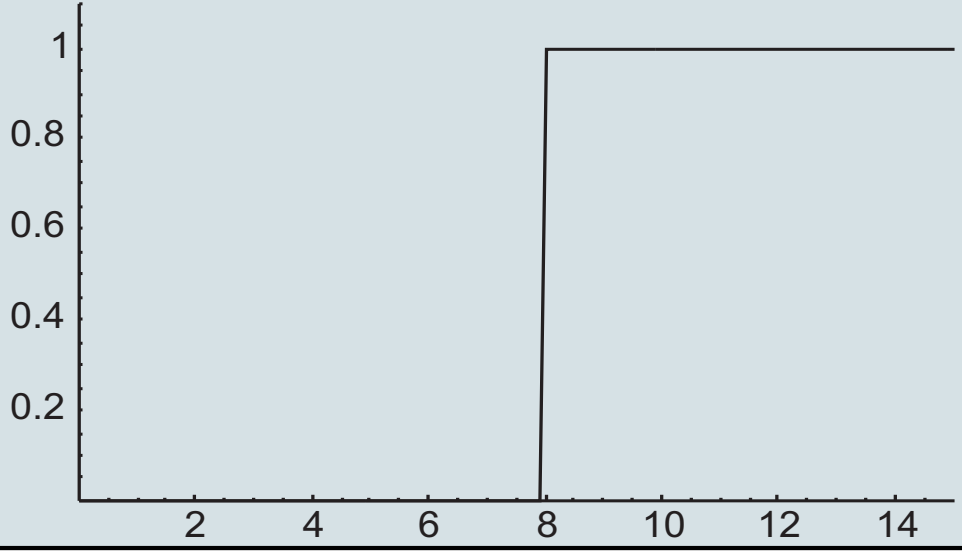
Here is an illuminating picture



All mass initially "lands" on the slope $y = \theta x$ and moves towards the south-west with direction $(1, 1/(\theta\rho))$.

How the system is crashing

- If the system starts empty, there is initially no impatience.
- When $z(t)$ reaches $1/\theta$, there is a sharp phase-transition: Suddenly, everybody becomes impatient.
- Holds for all service-time distributions!



The impact of variability

Other extreme: Users are either extremely patient or extremely impatient.

$\Theta = \epsilon$ with probability p and $\Theta = M$ with probability $1 - p$. In that case, the server utilization ρ_s can be as close to 1 as desired.

More variability in lead times has a positive effect on system performance.

In particular: more variability implies a higher service rate:
Compare two systems with identical λ, B but with different Θ_1 and Θ_2.

Proposition. If $\Theta_1 \overset{icx}{\geq} \Theta_2$, then $z_1 \leq z_2$.

Example II: Independent lead times

- We now assume that B and D are independent.
- We compare limiting values under different assumptions on the distributions.
- In all cases, $\rho = 1.5$, $\mathbf{E}[B] = \mathbf{E}[D] = 2$ and B and D either have an exponential distribution or a Pareto distribution with tail $(1+x)^{-1.5}$.

	B exp	B par
D exp	$z = 0.5000$	$z = 0.1174$
D par	$z = 0.2067$	$z = 0.0505$

More variability is always good!

Getting the time-dependent solution is possible

If D has an exponential with rate ν and $z(0) = 0$, then

$$z(t) = \lambda \int_0^t \mathrm{e}^{-\nu(t-s)} \mathbf{P}[B > \int_s^t \frac{1}{z(u)} \mathrm{d}u] \mathrm{d}s.$$

The solution is remarkably simple:

$$z(t) = (1 - \mathrm{e}^{-\nu t}) z.$$

In general, one can obtain $z(t)$ numerically by Picard-iteration.

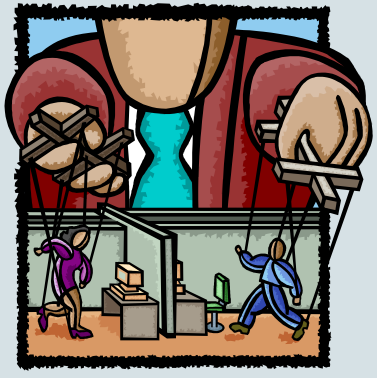
Performance analysis: summary

- Making customers more patient does not affect system performance in the long run.
- More variability leads to better system performance.
- Positive dependence between service times and lead times negatively affects system performance.
- Scenarios are possible in which almost all customers renege: The impact of reneging can be substantial.

Overview

- Modeling impatience in TCP networks.
- Performance Analysis.
- *A remedy: Admission control.*
- Reattempts.
- Summary.
- Related problems.

Controlling the impact of impatience



- To reduce the impact of impatience, one could perform admission control, i.e. bound the total number of customers in the system by some constant K.
- Trade off: customers may be blocked, but admitted customers are served at a higher rate, reducing the probability of reneging.
- Is it possible to improve system performance by admission control?

Admission control: Analysis

- Let q_K be fraction of customers that are admitted to the system.
- By Little's law, $z_K = \lambda q_K \mathbf{E}[\min\{z_K B, D\}]$.
- Observe that $q_K = 1$ if $z_K < K$. Consequently, $z_K = \min\{z, K\}$, with z the solution of the equation $z = \lambda \mathbf{E}[\min\{zB, D\}]$.
- If $z_K = K$, then q_K can be solved from the above equation for z_K.

Maximizing server utilization

- The fraction of successful customers is given by $V_K = q_K \mathbf{P}[z_K B < D]$.
- It can be shown that $V_K \to 1/\rho$ if $K \downarrow 0$ (small buffer). If the buffer is small, there is almost no reneging.
- This implies that the server utilization converges to 1 as $K \downarrow 0$.
- Hence, it makes sense to keep a small buffer in order to maximize the server utilization.

Maximizing user satisfaction

Things are not so clear when one aims to maximize the fraction of successful customers:

- When $D = \Theta B$, V_K is optimized by letting K become small.
- When D is constant and $\mathbf{P}[B > x] = \left(\frac{a}{a+x}\right)^b$, then V_K is maximized by performing no admission control at all ($K = \infty$).

Conclusion: Admission control increases the server utilization and sometimes also the fraction of successful transmissions.

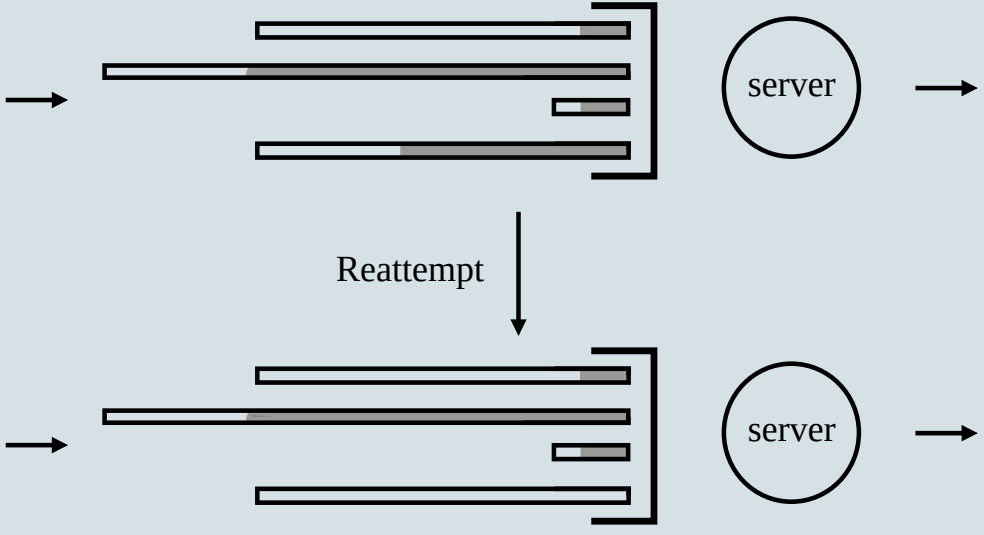
Overview

- Modeling impatience in TCP networks.
- Performance Analysis.
- A remedy: Admission control.
- ***Reattempts***.
- Summary.
- Related problems.

Extending the model to Reattempts

Typical user behavior: Impatient users of the Internet tend to click first on `STOP` and after that, immediately on `REFRESH`.

Assume that a customer, after having left the system due to impatience, retries immediately with probability $p \in (0, 1)$.



Reattempts cause bi-stability!

- The fixed-point equation becomes

$$z = \lambda \mathbf{E}[\min\{zB, D\}] + \frac{p}{1-p}\lambda \mathbf{P}[zB > D]\mathbf{E}[D \mid D < zB].$$

- Can have strictly positive solution, even if $\rho < 1$.

- Intuition: *the system is bi-stable*. For large, but finite r, the system can experience long periods during which there is a substantial reneging rate.

Summary and Conclusions

- The impact of impatience in overload can be substantial.
- More variability leads to better system performance.
- If the system is not overloaded, reattempting customers can have a significant impact.
- The impact of impatience can often be reduced by a simple admission control rule.

Overview

- Modeling impatience in TCP networks.
- Performance Analysis.
- A remedy: Admission control.
- Reattempts.
- Summary.
- ***Related problems***:
  1. Impact of scheduling on long sojourn times.
  2. Bandwidth sharing with heterogeneous flow sizes.

Impact of scheduling on long sojourn times

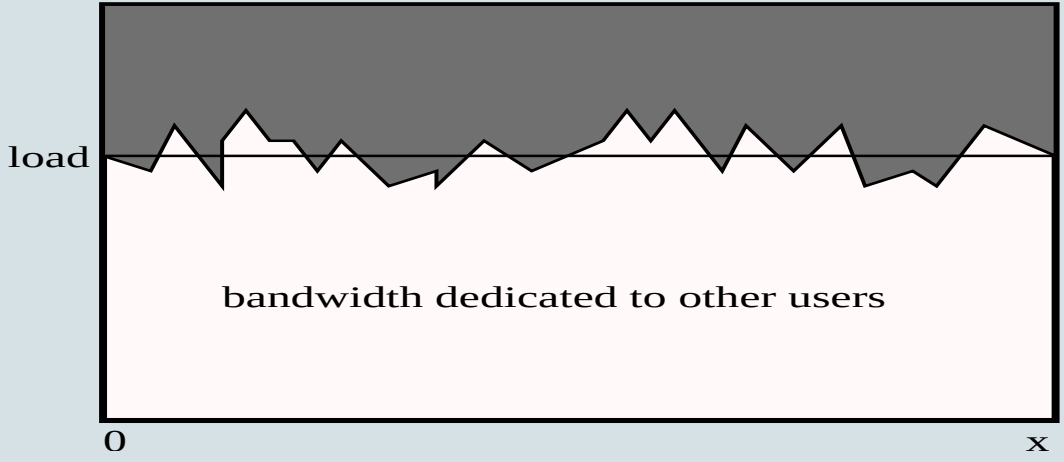
- Consider a system where several users share a common server.
- Service requirements are heavy-tailed: $\mathbf{P}[B > x] \approx x^{-\alpha}$.
- Which scheduling should one use? FIFO, or something more sophisticated?
- Usually, one compares average sojourn times.
- My research has focused on *the impact of scheduling on long sojourn times*

If you stay in the system for a long time...

... it's your own fault:

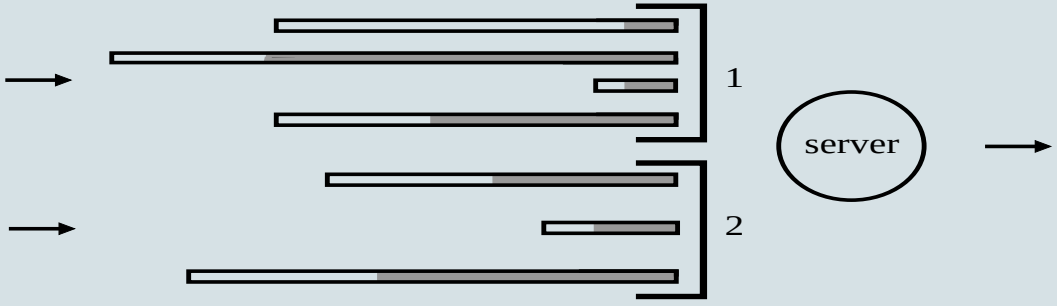
Zwart (ITC 1999), Zwart & Boxma (Questa 2000):

$$\mathbf{P}[V > x] = \mathbf{P}[B > x(1-\rho)](1+o(1)).$$



For FIFO: Long sojourn times are much more likely, and are caused other by another customer: NOT FAIR!

Bandwidth sharing with heterogeneous flow sizes



- Two classes of users share a link, all users receive the same service rate.
- Class 1 is well behaved: exponentially distributed service requirements.
- Class 2 is behaving badly: Heavy-tailed (Pareto) service requirements.

Question: Is class 1 well-protected from class 2?

Quality of Service for well-behaved users?

QoS for class 1 users: Large sojourn times should not happen too often.

It would be helpful if $\mathbf{P}[V_1 > x] \approx e^{-\gamma x}$.

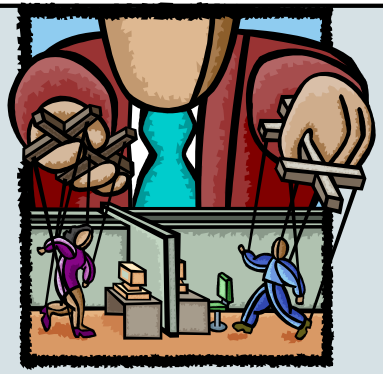
This would the case if class 2 is not present.

Borst/Nunez/Zwart (ITC2003): $\exists \delta > 0$:

$$\mathbf{P}[V_2 > x] \geq e^{-\delta \sqrt{x}}.$$

Users of class 2 have negative impact on QoS of class 1, so class 1 is NOT well-protected!

Solution: Admission control!



Upper bound the total number of users by $N < \infty$.

Then $\mathbf{P}[V_1 > x]$ has an exponential tail!

Important reason: In the system with blocking, there is a minimum guaranteed service rate: $1/N$, so

$$\text{Sojourn time} \leq N \times \text{service time}.$$